Master's thesis
Master's Programme in Data Science

Distribution Matching – Semi-Supervised Feature Selection for Biased Labelled Data

Moritz Johannes Lange

June 13, 2020

Supervisor(s): Kai Puolamäki

Examiner(s): Kai Puolamäki
Suyog Chandramouli

University of Helsinki
Faculty of Science

P. O. Box 68 (Pietari Kalmin katu 5)
00014 University of Helsinki

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Koulutusohjelma — Utbildningsprogram — Degree programme
Faculty of Science	Master's Programme in Data Science

Tekijä — Författare — Author

Moritz Johannes Lange

Työn nimi — Arbetets titel — Title

Distribution Matching – Semi-Supervised Feature Selection for Biased Labelled Data

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidantal — Number of pages
Master's thesis	June 13, 2020	49

Tiivistelmä — Referat — Abstract

In the context of data science and machine learning, feature selection is a widely used technique that focuses on reducing the dimensionality of a dataset. It is commonly used to improve model accuracy by preventing data redundancy and over-fitting, but can also be beneficial in applications such as data compression. The majority of feature selection techniques rely on labelled data. In many real-world scenarios, however, data is only partially labelled and thus requires so-called semi-supervised techniques, which can utilise both labelled and unlabelled data. While unlabelled data is often obtainable in abundance, labelled datasets are smaller and potentially biased. This thesis presents a method called distribution matching, which offers a way to do feature selection in a semi-supervised setup. Distribution matching is a wrapper method, which trains models to select features that best affect model accuracy. It addresses the problem of biased labelled data directly by incorporating unlabelled data into a cost function which approximates expected loss on unseen data. In experiments, the method is shown to successfully minimise the expected loss transparently on a synthetic dataset. Additionally, a comparison with related methods is performed on a more complex EMNIST dataset.

ACM Computing Classification System (CCS):
Information systems → Information retrieval → Document representation → Content analysis and feature selection
Computing methodologies → Machine learning → Machine learning algorithms → Feature selection
Computing methodologies → Machine learning → Learning settings → Semi-supervised learning settings

Avainsanat — Nyckelord — Keywords

Semi-supervised, Feature selection, Wrapper method, Bias

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — Övriga uppgifter — Additional information

Contents

1. Introduction

In data science and machine learning, datasets with many dimensions, or features, are a common occurrence. Text documents, for instance, can be represented by word counts. In such a dataset, the number of dimensions equals the size of the vocabulary [16]. Another example are datasets used in genetic analysis, where tens of thousands of genes are each represented by a feature [19]. Datasets of such high dimensionality often demand large storage and network bandwidth capacities. Additionally, training a model to perform a specific task – say, to predict the response to drug treatment from genetic data – might not require all available dimensions. Instead, in this scenario, only a small set of genes might determine the response to a specific drug. A model which uses all genes as features in training might even over-fit to the irrelevant genes.

These over-fitting and feature relevance issues can be addressed with feature selection [17], which chooses a subset of all features according to some criterion. In the case of drug response prediction, the criterion might seek to maximise the accuracy that a model can achieve with a certain number of features. Another common goal of feature selection is to reduce redundancy between the features of a dataset. By selecting meaningful features, human understanding of a dataset and its dimensions can be improved as well.

If feature selection involves training models, for example if it is used to optimise accuracy of a given model, then labelled training data is required. These feature selection methods are called wrappers [21]. However, real world datasets often provide only small samples of labelled data. The internet and databases, on the other hand, contain vast amounts of unlabelled data. This could for instance be unprocessed measurements, comments from social media, or scraped images from websites, or genetic data of people for whom medical records are not available.

Furthermore, a small labelled dataset is often less representative of the data generating distribution than the extensive unlabelled data. Humans who labelled the data might be biased, or chose a biased data sample to assign labels to. For drug response prediction, this might be a subset of patients in the full dataset that had a particular illness caused by specific genes, instead of a balanced set of patients with all possible illnesses. The labelled set might also simply be too small to represent a

balanced picture.

Feature selection which leverages both labelled and unlabelled data is called semi-supervised [29]. Especially in the case of biased labelled data, semi-supervised feature selection is expected to provide an advantage in optimizing model accuracy on unseen data, when compared to supervised feature selection which only utilises the biased labelled data.

Research into semi-supervised feature selection wrapper methods is scant. A recently published survey [23] lists only four methods of that kind ([1, 2, 22, 27]), which do not focus on distributional differences between labelled and unlabelled data.

This thesis looks to extend the field by presenting a new semi-supervised feature selection wrapper method, called distribution matching. The proposed approach in particular addresses the problem of labelled datasets which do not represent the generating distribution. It does so by weighing prediction losses of individual data points in a cost function, based on distances between labelled and unlabelled data points. As a result, the weighed labelled data used in the cost function approximates the distribution of the unlabelled data.

On a synthetic dataset, it is shown that distribution matching succeeds in minimizing the expected loss on unseen data for a selected feature subset and classifier. The proposed method is compared to similar methods on the EMNIST handwritten digit dataset [11].

After defining the problem statement tackled by distribution matching in the next section, relevant background theory is explained in Chapter 2. Chapter 3 introduces and explains the distribution matching method. Based on the definition, experiments are conducted in Chapter 4 to both analyse and compare the presented method to similar methods. The implications of these experimental results, as well as possible future work, are discussed in Chapter 5. Finally, Chapter 6 summarises the findings.

1.1 Problem statement

The objective of a feature selection method is to minimise expected error on unseen data instances. In a model development scenario, these unseen data instances are represented by an evaluation or hold-out dataset. If the model $f_j(x)$, trained on a selected subset of features S_j, tries to approximate the ground truth y of a data point (x, y), then the definition of the expected error, or expected loss, on unseen data is

$$\tilde{L} = \mathbb{E}_{(x,y)\sim G}\left(L(f_j(x), y)\right) . \tag{1.1}$$

Here, L is an arbitrary loss function that determines the individual loss between the prediction and ground truth of a data point. The expectation is calculated over data

points (x, y) that follow data generating distribution G. This distribution is the one that new unseen data points are expected to come from.

In the semi-supervised setting considered in this work, the presence of unlabelled and labelled data is assumed. Data points (x, y) in the evaluation dataset, as well as the unlabelled dataset, follow the generating distribution, while the labelled data does not necessarily do so. The problem that distribution matching tries to solve, is to minimise the expected loss over all possible S_j under these conditions where there may be distributional differences between labelled and unlabelled data.

2. Background

In this chapter, relevant concepts and related work are explained. The concept of feature selection is merged with the concept of semi-supervised learning to obtain semi-supervised feature selection.

To introduce a common notation for the remainder of the thesis, datasets will be represented as (x_i, y_i) pairs where $i \in [n]$ and $[n] = \{1, \ldots, n\}$. The data points are specified by x_i, which are m-dimensional vectors. Each element of the vector corresponds to one data dimension, or feature. The set of all m available features will be denoted by S.

Each data point has a corresponding label or ground truth y_i, which might not be known. If it is known, the data is called labelled, else unlabelled. In a modelling scenario, a model $f(x_i)$ tries to capture the relationship between x_i and y_i from a training dataset, and evaluates how well it succeeded on an evaluation dataset.

2.1 Feature selection

This section about the topic is in large parts based on the comprehensive reviews in [17] and [21]. Feature selection is the process of selecting a subset of features from a larger set of available features S. Let the subsets be denoted by S_j, where $j \in [p]$ enumerates the considered p feature subsets.

Feature selection provides various benefits for different situations. A smaller feature set in modelling leads to faster training and run times of models. Over-fitting can be avoided by removing redundant or correlated features. In the context of models, having fewer features might also improve understanding of models since they become less complex. If the model structure allows, the impact of each feature on the decision can even be extracted. Examples of these kinds of interpretable models include linear models and decision trees. Aside from modelling, albeit less commonly, feature selection can also be done with aims such as data compression or to gain understanding of the data without training models.

Mathematically, feature selection tries to optimise a loss function or cost function $c(S_j)$ when selecting S_j. Commonly used cost functions are information theoretic crite-

ria like the Akaike information criterion, or model performance measured by accuracy or residuals. The feature selection algorithm attempts to find a solution that is as close as possible to a global or at least local optimum of $c(S_j)$.

Feature selection reduces the dimensionality of data explicitly by selecting features, as opposed to other dimensionality reduction techniques which generally transform features as well.

There are three different categories of feature selection methods; filter methods, embedded methods, and wrapper methods. To which one a method belongs depends on whether it includes the training of models, and if the models select features as part of the training process.

Filter

Filter methods in feature selection are methods which analyse the data x_i without modelling the relationship to labels y_i. If labels are available, they can be used simply as additional information about each data point. Often filter methods are used as a data preprocessing step. According to some criterion, some features are selected and others discarded. For the most part, these criteria are based on how correlated or how similar features are. The goal is to minimise redundancy. The advantage of filter methods is that they are independent of a model and can prepare a dataset for later use even before the specific application scenario is determined.

Commonly, these methods aim to optimise a trade-off between relevance and redundancy of features, which is often based on correlation between features themselves and between features and class labels. A survey of filter methods, including the aforementioned, is presented in [8]. Other examples of filter methods select features or groups of features based on mutual information with the class labels, or based on different information-theoretic criteria such as the Akaike information criterion (see, e.g., [8] and [25]).

Embedded

Embedded methods do feature selection embedded in model training. This means that the objective function during model training is adjusted so that a model inherently prefers a certain feature subset. Often times this is done by regularisation. The advantage of embedded feature selection is that it does not require an extra step in addition to model training and is specific to the chosen model. Therefore it generally results in better model accuracy than a filter method. The disadvantage however is that a model designed for embedded feature selection needs to be available.

Common models that allow for embedded feature selection are those that allow

for a regularised loss function which punishes large amounts of features, such as LASSO linear regression [6]. Other examples of embedded feature selection are neural networks [20] and specifically regularised SVMs [10].

Wrapper

Wrapper feature selection methods wrap a feature selection method around the model training. This means that models $f_j(x)$ are trained for different feature subsets S_j. According to the cost function $c(S_j)$, calculated on a model instance, the best subset will be chosen. The advantage is, like with embedded methods, that the feature selection is optimised towards a particular model and hence often leads to better results than filter methods. Furthermore, this method does not require models which are adjusted for feature selection during the training phase, as is the case with embedded methods. Instead, feature selection can be wrapped around any kind of model without the need to adjust or understand the model. The disadvantage is that wrapper methods often require a model to be trained many times on different subsets S_j. For this reason they are generally the most computationally expensive variant of the three.

A wrapper feature selection method requires the definition of three required components:

1. A search method over the space of all feature subsets,

2. a cost function $c(S_j)$ indicating goodness of a feature subset, and

3. a machine learning model.

A search method is required since the number of possible feature subsets is 2^m if there are m features. Hence the amount of models which would need to be trained quickly becomes too big for investigating the entire space. The search for the globally best feature subset is an NP-hard problem [5] and therefore needs to be approximated for larger m.

The model and an adequate cost function to measure its performance can be chosen freely depending on the requirements and provided data. A common performance measure would for example be the accuracy of a model trained with feature set S_j.

2.1.1 Relevant work

Over the last decades, many search methods have been developed for feature selection wrappers. One of the most well-known wrapper methods is greedy forward selection [9].

Greedy forward selection

Greedy forward selection is one of the simplest and fastest search algorithms for wrapper methods. For a subset S_j of features, the algorithm tries to add each remaining feature from S that is not yet in S_j. The feature providing the best performance gain is kept as an addition to the subset. Starting from an empty set, the subset is thus grown one by one, until a stopping criterion is reached. Schematically, the algorithm works as follows:

1. Start with empty set of features $S_j = \{\}$
2. $\forall s \in S \backslash S_j$ create new sets $S_j^\star \leftarrow S_j \cup s$

   Train model $f_j^\star(x)$ with features in $S_j^\star$

   Calculate $c(S_j^\star)$ on $f_j^\star(x)$
3. Keep the best performing $S_j^\star$ as new S_j: $S_j \leftarrow \text{argmin}_{S_j^\star}\left(c(S_j^\star)\right)$
4. Stop if stopping criterion is fulfilled, otherwise proceed with step 2 using the new S_j

The stopping criterion commonly is either a fixed number of features required for S_j, or if $c(S_j)$ starts to rise again from the previous step. That would mean the addition of the last feature has led to decreased model performance.

Greedy forward selection is an efficient search algorithm since it has a polynomial complexity, conditional on how the stopping criterion depends on n. In practice, the size of the final feature set grows slower than n, hence the degree of the polynomial is between 1 and 2.

Due to its simplicity and efficiency in selecting features [9], greedy forward selection remains a widely applied method and shall also be the search algorithm which is used in this thesis for distribution matching.

Related methods

A noteworthy related approach is greedy backward elimination, which follows the same principles but starts from the full set of features S, and eliminates individual features until a stopping criterion is reached. A hybrid approach that can add or remove features to a given subset, depending on which has the best effect on accuracy, is also known. Compared to greedy forward selection this has the advantage that features can be removed again later, if redundancies or new interactions emerge. Similarly, unlike with greedy backward elimination, features can be added again. These advantages, however, come at the price of a more extensive and complex search strategy. Both greedy

backward elimination, and the hybrid approach, are also discussed in [9]. Beyond these, [21] debates a range of alternative search methods with varying complexity, which are however not used by methods discussed in this thesis.

2.2 Semi-supervised learning

The cost function proposed in this thesis for distribution matching relies on labelled data, as well as on unlabelled data. Using both these types of data is referred to as semi-supervised learning. The n data points are divided into n_l points with known labels y, and n_u points where y is unknown.

Semi-supervised learning is one of three learning paradigms, and extends the first two – unsupervised and supervised learning – to deal with the mix of labelled and unlabelled points by using their distributional information. Some sources define additional paradigms to these three, but those are beyond the scope of this thesis. For an in-depth review of the differences, the introduction paper [29] by Zhu et al. can be recommended. The same author has also written a literature survey on semi-supervised learning [30].

Unsupervised learning

Unsupervised learning describes all algorithms that do not require labels y. These algorithms commonly learn properties such as distribution or structure of a dataset instead of modelling it. The two most prominent applications are dimensionality reduction and clustering. In dimensionality reduction, a dataset with n data points and m_1 features is transformed so that the same n data points are described by only $m_2 < m_1$ features. In clustering, data points are assigned to groups so that points within a group behave in a similar way. Unsupervised learning approaches can help with understanding a dataset and its distribution, or with reducing data to a smaller set that is easier to handle.

Supervised learning

Supervised learning describes algorithms that utilise the full tuple (x, y), usually by modelling the dependence between x and y with a model $f(x)$. These learning algorithms can span from simple linear regression to highly complex neural networks [12]. Over the last several decades, plenty of model types ranging from decision tree models to Support Vector Machines (SVM) [24] have been developed for various kinds and distributions of data. Each model makes its own assumptions about the data. Supervised learning is used to understand a relationship between x and y, as well as to predict the

labels for future data points.

Semi-supervised learning

With partly labelled data, the labelled subset of points can be used for supervised learning as described above. The unlabelled data points are not useful for supervised learning directly, but can still provide insights about properties of the data, such as data distribution, by means of unsupervised learning. The insights into properties of data points x gained with unsupervised learning is of higher quality when using both labelled and unlabelled data, compared to just using one or the other. Those properties, such as data distributions, can be used for improving the quality of the supervised approach for the n_l labelled data points in turn.

2.3 Semi-supervised feature selection

Semi-supervised feature selection is feature selection done in a semi-supervised setup. In a scenario where unlabelled and labelled data is available, both should be used not just for model training, but also for feature selection, to obtain the best results.

The available labels, for a subset of the data points, can be used both as additional information about those points, or to train models. Therefore, all three feature selection categories presented in Section 2.1 can be applied. The same advantages and disadvantages of each also apply here, although the efficiency of utilizing all information contained in both labelled and unlabelled data determines how well a feature selection method performs. For labelled data this kind of information would primarily be the ground truth, and for unlabelled data it would be the distribution of data points.

For a wrapper approach like the one presented in this thesis, there are two possibilities to integrate unlabelled data points into the usually supervised setup:

1. In the cost function $c(S_j)$
2. In the models $f_j(x)$

The cost function can include information from unlabelled data by introducing terms calculated on it, in addition to a performance measure of a supervised model that is only based on labelled data points. These terms could for example be weights or penalties based on the data distribution. Another option is to predict labels for the unlabelled data points using a model trained on labelled data points, and thus providing a ground truth for the full $n = n_l + n_u$ data points.

The model training can be adjusted by using a semi-supervised model training approach. Since the wrapper method works independent of the model it is wrapped

around, this approach does not alter the feature selection method itself.

2.3.1 Relevant work

The research that has been done about semi-supervised feature selection is nicely summarised in [23]. This section is a summary of their findings, with a particular focus on the wrapper methods related to the approach presented in this thesis.

Filter Most semi-supervised feature selection methods are in fact filter methods that do not rely on models. They mostly construct graphs of data points, encoding the structure in labelled and unlabelled data. This is done by defining distances between data points (see, e.g., [14, 15, 28]). Labels for unlabelled data points can be inferred from labelled data points in the same part of the structure. Features in graph-based methods are commonly selected independently of the labels, by testing which features best preserve graph geometry.

Embedded Embedded feature selection methods in the semi-supervised setup often use sparse models, which inherently ignore certain features, after labelling the unlabelled data. This effectively creates a supervised setup. Another approach entails the use of semi-supervised support vector machines (S^3VM) which are specifically built for dealing with semi-supervised data and select features by enforcing sparsity as well [3].

Wrapper For wrapper methods in semi-supervised feature selection there is the least amount of previous research, according to [23]. The wrapper methods that exists are based on one or more models, which are trained on the labelled data and expand the labelled dataset by predicting labels for the unlabelled data (see [1, 2, 22, 27]). In the case of multiple models, the label with the highest confidence gets assigned to a data point. Feature selection is then done with standard wrapper methods for labelled data, like forward selection.

From a semi-supervised perspective, these methods utilise both labelled and unlabelled data with three different methods:

Graphs With graphs, relationship between data points is encoded as geometric structure. Labels for unlabelled data points are inferred from closeby labelled points.

Self-training In self-training [2] and co-training [1], models are trained on the labelled data points and the set of labelled points is extended by predicting labels on unlabelled data points.

Semi-supervised model A semi-supervised model internally can use both labelled and unlabelled data points for training. An example is the S^3VM.

Of the semi-supervised feature selection methods presented in [23], two in particular have similarities with distribution matching. The locality sensitive discriminant feature method, detailed in [28], is similar since it also utilises a form of distance measures between data items. Forward semi-supervised feature selection, which is presented in [22], is similar in that it uses a wrapper method based on greedy forward selection.

Locality sensitive discriminant feature

Locality sensitive discriminant feature (LSDF) is a semi-supervised feature selection filter method for classification tasks, presented in [28]. The algorithm creates graphs from data points and ranks features according to how well they preserve local graph structure. The survey [23] has mixed results for performance of semi-supervised feature selection methods, depending on the dataset. However, because this method still seems to be one of the better performing methods, and because it uses a form of distance score between data points for constructing the graphs, it will, in Chapter 4, be compared against distribution matching.

Specifically, LSDF constructs two graphs with data points as vertices. The first is a within-class graph, which has edges between labelled data points with the same label. It also contains edges between data points if they are K-nearest neighbours of each other and at least one is unlabelled. These edges, however, have a much lower weight, which is independent of the actual distance. The second graph is a between-class graph, which connects all labelled data points that have different labels.

For the within-class and between-class graph, the respective Laplacian matrices are computed. A Laplacian matrix encodes the edge connections of the graph, and hence the graph structure, in form of a matrix. From the Laplacian matrices, a feature importance score that represents how well each feature preserves the structure of both graphs.

Forward semi-supervised feature selection

The forward semi-supervised feature selection wrapper method proposed in [22] is based on self-training. The underlying concept is to label random subsets of the unlabelled data by classifiers trained on the labelled data, and then perform greedy forward selection to find the best features.

This method is one of only four wrapper methods presented in [23], and is in Section 4 also compared to distribution matching. Like the implementation of distribution matching analysed in this thesis, forward semi-supervised feature selection uses

greedy forward selection as a search method. The authors presenting forward semi-supervised feature selection claim specifically that their method is suited for biased labelled datasets, rendering it a particularly fitting comparison.

Forward semi-supervised feature selection starts by finding an initial set of features using normal greedy forward selection, as described in Section 2.1.1, on the labelled data. Then, the labelled data is extended by labelling some random part of the unlabelled data. This step is repeated several times, and each time greedy forward feature selection selects a subset of features on the extended labelled datasets. The procedure results in several feature sets, out of which the most frequent feature is finally added to the initial set of features. That whole process is repeated several times, until the initial set of features has grown to a predefined size. This final set is returned as the subset of selected features.

3. Distribution matching

In this chapter a new semi-supervised feature selection method, called distribution matching (DM), is presented. It is characterised by a cost function which utilises weights that are based on distances between labelled and unlabelled data points.

3.1 Motivation

In semi-supervised setups, labelled data is often much scarcer than unlabelled data, which commonly can be scraped from the internet or databases in abundance. Unlabelled data, when available in large quantities, is expected to approximately follow the data generating distribution [13]. Labelled data, on the other hand, has undergone a procedure where adequate labels are assigned to each data point, often by hand. This is often the reason for the small size of labelled datasets. It potentially introduces bias, because the labelled dataset might not be of sufficient size to capture the generating distribution, or only a biased sub-sample might have been labelled. In such cases, the distribution of the labelled data is different to that of the generating distribution.

Conventional supervised feature selection methods such as greedy forward selection would consequently train models on biased data, which can be expected to negatively influence performance $\tilde{L}$ on evaluation data [18].

As discussed in the previous chapter, not much research on semi-supervised feature selection methods of the wrapper type has been conducted yet. The papers that exist focus on extending the labelled dataset by labelling unlabelled data (see, e.g. [1, 2, 22, 27]). That approach does not explicitly address the bias. Instead, the focus is on enlarging the training dataset to gain better accuracies. Filter type methods, of which there are more instances, are not optimised towards a given model, and embedded methods are limited to certain types of models.

The DM method proposed here, in contrast, provides a wrapper method that does feature selection tailored to any given model in a semi-supervised setup, handling bias in the labelled data explicitly by weighing it according to the distribution of the unlabelled data. This approximates the expected loss $\tilde{L}$ on unseen data directly.

3.2 Theory

Distribution matching weighs the individual loss of each labelled data point in the overall cost function $c(S_j)$ such that the distribution of unlabelled data points is approximated. Any search function or model type can in principle be used with distribution matching. However, in this study the search algorithm is chosen to be greedy forward selection due to its simplicity, efficiency and widespread use [17].

The cost function approximates $\tilde{L}$, as described in Equation (3.2), by scaling each individual loss of a labelled data point x_i with a weight w_i. It reads

$$c(S_j) = \frac{1}{n_l} \sum_{i \in [n_l]} w_i L(\hat{y}_i^j, y_i) \,, \tag{3.1}$$

where $L(\hat{y}_i^j, y_i)$ is the loss between the true label y_i of data point x_i and the corresponding prediction $\hat{y}_i^j$. The prediction is obtained with a model $f_j(x_i)$ that was trained on feature subset S_j. The loss $L(\hat{y}_i^j, y_i)$ can be freely defined; a common choice for a supervised loss function is the ℓ_1 or ℓ_2 norm of $\hat{y}_i^j - y_i$ in regression tasks, or accuracy in classification tasks.

With appropriately chosen weights, $c(S_j)$ approximates $\tilde{L}$ as

$$\frac{1}{n_l} \sum_{i \in [n_l]} w_i L(\hat{y}_i^j, y_i) \approx \mathbb{E}_{(x,y)\sim G} \left(L(f_j(x), y) \right) \,, \tag{3.2}$$

since the weights emphasise those data points which are similar to the unlabelled distribution, and thus, in this setting, characteristic for G. The cost function $c(S_j)$ approaches $\tilde{L}$, if $n_l \to \infty$ and w_i weigh the data points such that they represent G, assuming that all data categories from G are present in the labelled set.

For this purpose, the weight w_i for a labelled data point x_i is computed as the softmax distribution of the distances between labelled and unlabelled data points, averaged over all unlabelled points. Consequently, the more and the closer unlabelled data points are around labelled data point x_i, the larger w_i. It is defined as

$$w_i = \frac{1}{n_u} \sum_{o \in [n_u]} \frac{\exp(-\beta \cdot d_x(x_i, x_{n_l+o}))}{\sum_{q \in [n_l]} \exp(-\beta \cdot d_x(x_q, x_{n_l+o}))} \,, \tag{3.3}$$

assuming that n_l labelled data points are followed by n_u unlabelled data points with continuous indexing. This definition relies on a distance measure $d_x(x_i, x_l)$ between data points x_i and x_l. Possible definitions for d_x, based on predictions of models trained on various feature subsets, are presented in Section 3.3.

The factor β controls how strongly the influence decays with distance. For $\beta \to \infty$, only the nearest neighbour in the unlabelled set affects the weight. Large

β generally reduce many weights to such small values, that only few w_i remain significant. For $\beta = 0$, the distances don't affect the weights and hence all w_i have the same value $1/n_l$. This means that for $\beta = 0$, DM becomes standard greedy forward selection which does not utilise unlabelled data at all.

3.3 Data distance measures

The distance between data points is commonly calculated by using an ℓ_p-norm, generally ℓ_1 or ℓ_2, between the data points. However, in datasets with large numbers of features, as they are considered here, those distances tend to all have similar values. This phenomenon is known as distance concentration [4]. Additionally, every feature has a similar influence on the distance, regardless of whether it is useful for modelling.

To address these issues, this thesis uses distance measures d_x which are based on a prediction vector, generated from several models trained on randomly selected features. Specifically, the distance $d_x(x_i, x_l)$ between two data points x_i and x_l is calculated from predictions by several models $f_j(x_i)$ trained on different feature subsets S_j. This results in a vector of predictions $\hat{y}_i$ for each data point, with elements indexed by $j \in [p]$, $[p] = \{1, \ldots, p\}$. The length p can be controlled so as to avoid distance concentration.

Each subset S_j is sampled randomly from S with a suitable size k, which needs to be selected depending on the dataset. The randomly sampled subsets need to result in models of at least mediocre accuracy. Models with no predictive power would lead to noise in the predicted vectors $\hat{y}_i$, because predictions would be independent of the data point, and hence contains no relevant information. However, similar to the feature selection problem it is generally impossible to search the entire space of possible subsets of S for suitable candidates. For this reason, random sampling is used. Alternative, more involved approaches to random sampling do also exist and are discussed further in Chapter 5.

In the scope of this thesis, two prediction-based distance measures are used. Additionally, a data-based ℓ_1-norm distance is used as a baseline to compare against.

Definition 3.1. The *correlation distance* between two data points is defined as

$$d_x^{\text{cor}}(x_i, x_l) = 1 - |\text{cor}(\hat{y}_i, \hat{y}_l)|\,, \tag{3.4}$$

where $\text{cor}(\cdot)$ is the Pearson correlation coefficient. The vector $\hat{y}_i$ contains all p predictions $\hat{y}_i^j$, for data point x_i obtained with the p different feature subsets.

The correlation distance is smaller the stronger x_i and x_l are correlated. The second measure is the minres distance, which only works for discrete classes.

Definition 3.2. The *minres distance* between two data points with a set of possible class labels C is defined as

$$d_x^{\text{minres}}(x_i, x_l) = \min_{c_i, c_l} \left(\frac{1}{p} \sum_{j=1}^{p} \left| \left| \hat{y}_i^j - c_i \right| - \left| \hat{y}_l^j - c_l \right| \right| \right) , \tag{3.5}$$

where $c_i \in C$ and $c_l \in C$ are all possible class labels.

The minres distance finds for both data points the class labels which lead to the smallest possible residuals. Assuming these found class labels are the ground truth, it calculates the mean difference between the residuals. In a binary classification setup, $C = \{0, 1\}$ allows for d_x^{minres} to work also for vectors $\hat{y}_i$ of continuous class probabilities. A possible weakness of d_x^{minres} is that if predictions are constantly close to a different class label than the true label, the distance will become small as well.

Definition 3.3. The ℓ_1-norm distance is a common distance measure for data points, based directly on the data values. It is used here as comparison to the prediction-based distances d_x^{minres} and d_x^{cor}, and is defined as

$$d_x^{data}(x_i, x_l) = ||x_i - x_l||_1 . \tag{3.6}$$

4. Experiments

In this chapter, distribution matching is analysed experimentally, and evaluated against standard greedy forward selection, LSDF and the forward semi-supervised feature selection method described in Section 2.3.1.

While distribution matching, as defined in Equation (3.1), allows solving multi-class and regression problems, LSDF is limited to classification tasks. In this thesis all experiments are conducted for binary classification tasks, since the correlation distance the way it is currently defined does not support multi-class problems.

All experiments are conducted in R, version 3.6.3. The exact parameters that are used in all experiments, are listed in Table B.1. Instead of $c(S_j)$, the experiments report accuracy as $1 - c(S_j)$. This makes results more readily understandable.

4.1 Datasets

In all experiments, two datasets are used. The first is a synthetic dataset, the second is the EMNIST dataset [11]. To obtain a semi-supervised setup with labelled data that is differently distributed from unlabelled data, several subsets are sampled from each dataset according to distributions specified in Table 4.1.

For feature selection, a labelled and unlabelled dataset are generated. Evaluation of performance of selected feature sets is done with an evaluation training set and an evaluation testing set. All performance results reported in the thesis are calculated on the evaluation sets. There is no overlap between labelled, unlabelled, and either of the evaluation sets.

Each experiment is conducted five times for five sub-samples of the datasets with different random seeds. Reported accuracies are the average over those five runs, and error bars report the standard deviation.

Synthetic dataset

A simple and easily understandable synthetic dataset, described in Table 4.2, is used since it provides understandability and transparency. The task is to determine a binary

Table 4.1: Data preparation parameters. From EMNIST, samples are drawn from digits 3, 5, 1 with the shown distribution (same order). For the synthetic data, samples are drawn from categories one and two with the shown distribution (same order, for categories see Table 4.2). For EMNIST, the number of labelled data points n_l depends on the experiment.

	EMNIST		Synthetic data	
	Distribution	N	Distribution	N
Labelled	(0.5, 0.4, 0.1)	n_l	(0.7, 0.3)	200
Unlabelled	(0.5, 0.1, 0.4)	$1000 - n_l$	(0.3, 0.7)	500
Evaluation train	(0.5, 0.1, 0.4)	1000	(0.3, 0.7)	500
Evaluation test	(0.5, 0.1, 0.4)	1000	(0.3, 0.7)	500

class label from two features. The classifier is a decision rule. For data point x, with any number of available features, determine $\hat{y}$ with

$$f(x) = \begin{cases} 1 & \text{if } x \text{ contains a 1} \\ 0 & \text{otherwise}\,. \end{cases} \tag{4.1}$$

The three data points in the synthetic set are split into two categories. The first category contains points $x = (1, 0)$ and $x = (0, 0)$. In this category, the feature $F1$ determines the class c. In category two, which consists of points $x = (0, 1)$ and $x = (0, 0)$, feature $F2$ determines the class. The point $x = (0, 0)$ appears in both categories.

Table 4.2: The synthetic dataset is comprised of two features F1 and F2. It has three possible data points. The label of each point x_i, specified through $F1$ and $F2$, is represented by class c.

$F1$	$F2$	c
1	0	1
0	1	1
0	0	0

EMNIST

EMNIST is a dataset of handwritten digits, each of which has 784 pixels or features [11]. It is used because it provides more complex data with many features, while allowing for straightforward adjustment of the distribution of digits. The classification task in all experiments is to determine whether a digit is a 3. For such binary task, only three digits are necessary to implement a distributional shift between labelled and

unlabelled data. Therefore EMNIST is preprocessed to contain only digits 1, 3 and 5. Pixels, or features, which have a variance of 0 across all data points are removed. This brings the number of features down from 784 to a bit over 600, with the exact number depending on the random selection of data points.

4.2 Distribution matching

First, the distribution matching method is analysed in detail. The relevant components are the distance measure d_x and the parameter β, which both are used to calculate weights w_i.

The implementation of distribution matching, as defined in Section 3, requires adjustments to make it numerically stable for large β. Division by zero is avoided by using the log-sum-exp trick in the calculation of w_i [7]. This numerical adjustments is used for all experiments described in this chapter. Additionally, for calculating distance measures d_x, classifiers return probabilities instead of class labels, since they contain additional information on the confidence of the classifier. The distance measures defined in the previous chapter work with probabilities in a binary classification setting.

4.2.1 Influence of d_x

Distribution matching inherently assumes that the distance measure $d_x(x_i, x_l)$ is able to adequately represent how similar two data points are. With data distributions as in Table 4.1, the average distances to the unlabelled data for each data category should depend on how many similar data points are present in the unlabelled data. This is because similar points should have a small distance, and the amount of similar data points affects that average. Specifically, in the synthetic data, points $(0, 1)$ are more prevalent in the unlabelled data and should therefore have smaller average distance than points $(1, 0)$ which are less prevalent in the unlabelled set. The weights of points $(0, 1)$ is hence expected to be higher. In the EMNIST case following a similar argument, the digit 1 should have smaller average distances and larger weights than the digit 5.

This experiment investigates whether the assumption holds for all distance measures d_x^{minres}, d_x^{cor} and d_x^{data}. For each of the distance measures, average distances and also weights are calculated for the labelled data point. Predicted vectors $\hat{y}$ are calculated as class probabilities, using the optimal classifier in Equation (4.1) for the synthetic dataset, and a Support Vector Machine (SVM) for EMNIST. The detailed parameters are listed in Table B.1 for distribution matching.

Table 4.3: The mean distances and weights calculated for the synthetic dataset, for different distance measures. Values are rounded to two significant digits. Not the different magnitude of weights across different data points. Mean distance refers to the average distance of a labelled data point to all unlabelled data points. Due to the simplistic nature of the synthetic dataset, all points in a data category share the same mean distance and weight. In all cases, $n_l = 200$, $n_u = 500$, $\log \beta = 5$.

Data		d_x^{minres}		d_x^{cor}		d_x^{data}	
		Mean distance	Weight	Mean distance	Weight	Mean distance	Weight
1	0	0.370	0.0022	0.75	0.0047	0.59	0.0022
0	1	0.130	0.0120	0.62	0.0140	0.41	0.0120
0	0	0.098	0.0052	1	0.0025	0.25	0.0052

Synthetic dataset

The average distances for the synthetic data, as well as weights computed from the distances, are listed in Table 4.3. For all measures d_x^{minres}, d_x^{data} and d_x^{cor}, the distances follow the expected pattern. Average distances of d_x^{minres} are in fact a scaled version of average d_x^{data}, but this is only coincidence due to the random seed and subsequent selection of S_j used to calculate d_x^{minres}. The weights calculated from all distance measures follow the expected pattern as well, with (0,1) having the largest weight.

The conclusion of this experiment is that all distance measures work as expected with the synthetic data, and lead to appropriate weights.

EMNIST dataset

The histograms of average distances, as well as weights, for the three EMNIST data digits are shown in Figure A.1. For all three distance measures, the average distances of digit 1 are clearly smaller, and weights larger, when compared to digit 5. It is however noteworthy that distributions overlap, showing that distances and weights between two different digits on average follow the expected pattern, but in individual cases might violate it. As with the synthetic data, it can be concluded that all distance measures worked as expected.

Generally, from the results of this experiment both on the synthetic as well as the EMNIST dataset, and from the results of the experiment presented in the next section, none of the distance measures can be said to perform better or worse than its competitors. The only exception is when d_x^{cor}, as shown in Figure 4.1 in the next section, does not succeed in approximating the theoretically expected accuracy on synthetic evaluation data, which might be due to the limited information contained in

the predicted labels for subsets S_j for synthetic data in particular.

4.2.2 Influence of β

The second parameter in w_i is β. On a conceptual level, β controls how fast the influence of unlabelled data points on a labelled data point decays with distance. Practically, since it is a factor scaling the term that is exponentiated, calculated distance values will be pulled apart by large β. This is because smaller distances are magnified disproportionately after exponentiation. If β is small, on the other hand, exponentiation will not be much different to linear transformation of the distances. Hence, for small β, the distances will stay in proportion to each other.

The question addressed by this experiment is how the scale of β affects the performance of the presented DM method. This is investigated for all three distance measures.

The experiment is again conducted on both the synthetic dataset and the EMNIST dataset. The full feature selection algorithm, including the DM cost function $c(S_j)$ defined in Equation (3.1), and greedy forward selection, is run for different β values. An SVM is used for calculating both distance measures, as well as predictions during greedy forward selection. Parameters can be seen in Table B.1.

Synthetic dataset

For the synthetic dataset, it is expected that DM selects $F2$ as the first feature, since it is the most prevalent in the unlabelled dataset. Given the distributions from Table 4.1, for $N \to \infty$, expected accuracy on evaluation data is 0.65 when selecting $F1$, and 0.85 when selecting $F2$.

For all three distance measures, the experiment shows that expected accuracy on evaluation data, for β below a threshold, is larger for $F1$ than for $F2$. With increasing β, the expected accuracies move towards each other. Only when β surpasses a threshold, which lies between $0 < \log \beta < 2$, does the expected accuracy become higher for $F2$, and DM consequently chooses $F2$. This result holds for all three distance measures, as can be seen in Figure 4.1. However, for d_x^{data} and d_x^{minres}, the theoretically expected values 0.65 and 0.85 are correctly approximated, while d_x^{cor} returns different expected accuracies for $\log \beta > 0$.

The reason why DM wrongly prefers $F1$ for β below a threshold is that for too low β the weights of data points $(0, 1)$ are not yet high enough in relation to weights for points $(1, 0)$. In other words, the influence of unlabelled data points does not decay strongly enough with distance, so that unlabelled points $(1, 0)$ influence the weights of labelled points $(0, 1)$ too much. The fact that d_x^{cor} does not correctly approximate the

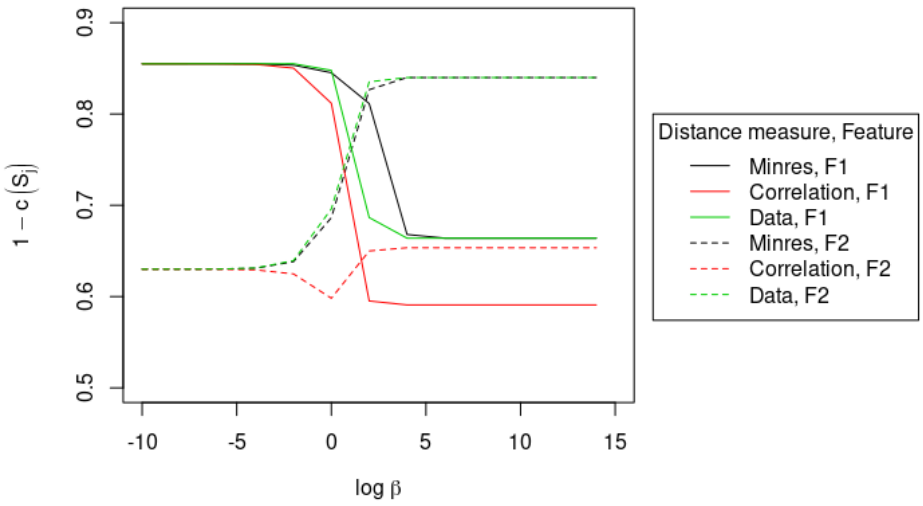


Figure 4.1: Expected accuracy $1 - c(S_j)$ on evaluation data, for features $F1$ and $F2$ of the synthetic dataset and all different distance measures. In this experiment, $n_l = 200$, $n_u = 500$, and $\log \beta = 5$.

theoretically expected values of 0.65 and 0.85 for $\log \beta > 0$ is likely caused by too little and too restricted feature subsets S_j that d_x^{cor} is calculated on.

The conclusion of this experiment is that β can be selected too small, resulting in data points of different categories influencing the weights of each other too much. When selecting β large enough, this influence is reduced to an amount that allows DM to choose the feature that maximises expected accuracy on evaluation data.

EMNIST dataset

For the EMNIST dataset, it makes more sense to compare achieved accuracies than which of over 600 features is chosen. The accuracies achieved on evaluation data with the first 6 selected features of DM, for different values of β, is shown in Figure 4.2. The threshold of first six selected features is chosen, because fewer features have stronger varying accuracies, and for more than six features accuracies are mostly converged to their maximum, at which point the selection of features becomes less meaningful. Both of this is visible in Figure A.2 in the attachment.

Figure 4.2 shows that accuracies within the reported standard deviations do not significantly change between different β values or distance measures. There are however, slightly lower average accuracies for the lowest β value, albeit with a large uncertainty. Uncertainty also increases for the highest β value.

The result of this experiment for EMNIST data is that with EMNIST, the choice of β does not seem to be critical within reasonable bounds. The drop for the lowest

reported β might be due to the same phenomenon that was reported for the synthetic dataset. For the largest reported β value, the point might be reached where enough weights become negligible so that the predictive power of $c(S_j)$ starts being reduced. This is a possible explanation for an increase in uncertainty.

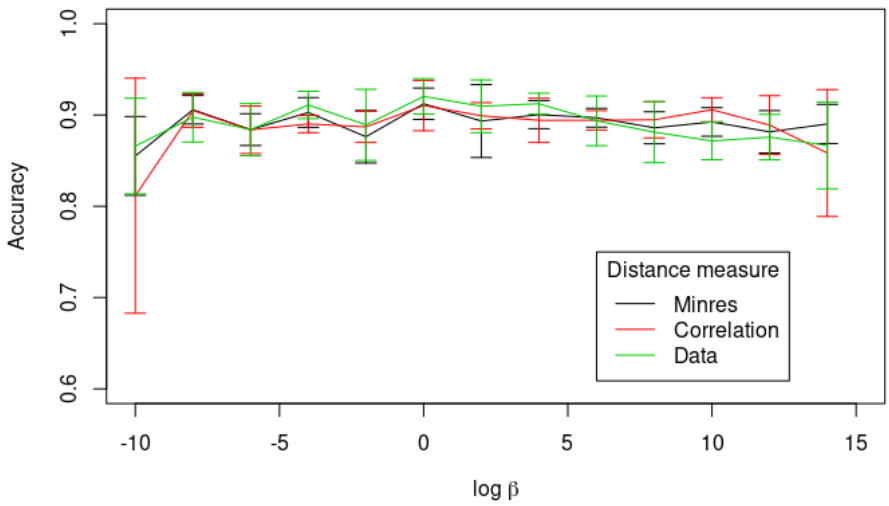


Figure 4.2: Best performance that different β can achieve for EMNIST with six features, for different distance measures. The reported accuracy is the mean over five different random seeds, and the error bars indicate the standard deviation. In this experiment, $n_l = 300$ and $n_u = 700$.

4.3 Method comparison

Distribution matching is in this section compared to the LSDF and forward semi-supervised feature selection methods presented in Section 2.3.1, as well as the supervised greedy forward selection.

The purpose of the proposed distribution matching method is to minimise the expected loss $\tilde{L}$ over unseen data, given a biased labelled dataset. This experiment seeks to investigate how DM performs on this measure, when compared to the related methods.

Consequently, the experiment runs each method for the distributions specified in Table 4.1, and compares the resulting accuracies on evaluation datasets. The cost function of distribution matching is, as before, combined with greedy forward selection. The parameters used with each method are listed in Table B.1. Experiments are run for both the synthetic dataset and the EMNIST dataset. The classifier used with the synthetic data is the optimal classifier defined in Equation (4.1), and the classifier used with EMNIST is again the SVM. The experiments on the proposed DM method do

not consistently prove any of the available distance measures superior for the provided datasets. However, since d_x^{data} is in principle more susceptible to the effect of distance concentration, and d_x^{cor} did not always return theoretically expected values in Section 4.2.2, d_x^{minres} is chosen to be used with distribution matching in this experiment.

4.3.1 Synthetic dataset

As discussed in the previous section, selecting feature $F2$, for $N \to \infty$, results in an accuracy of 0.85 on evaluation data while selecting feature $F1$ only gains an accuracy of 0.65. Hence, $F2$ needs to be selected in order to maximise expected accuracy on the evaluation dataset. Distribution matching selects $F2$, as shown in Section 4.2.2, if β is chosen sufficiently.

Greedy forward selection, without the weights used in distribution matching, hence always selects feature $F1$ from the labelled data, since it is more prevalent and hence better at predicting labels.

LSDF with default parameters selects F1, assigning it a significantly higher score than F2 (0.033 and 0.013, respectively). The failure of LSDF to select feature $F2$ might be explained by its difficulty to efficiently utilise K-nearest neighbour, since the distances are discrete. In that case, the unlabelled data remains largely inaccessible for LSDF.

The forward semi-supervised feature selection method, finally, is not applicable to the synthetic dataset since it first selects a number of features with greedy forward selection. Its result is therefore the same as that of greedy forward selection for the first feature.

The underlying principle of forward semi-supervised feature selection, however, is to label a part of the unlabelled data and generate an extended labelled set. With the optimal classifier from Equation (4.1), using both features for classification, the majority category in the extended labelled set, obtained with such a method, depends on how much of the unlabelled data is labelled. When enough of the differently distributed unlabelled data points become labelled, data point $(0, 1)$ will become more prevalent than data points $(1, 0)$, and this approach will select feature $F2$.

In conclusion, both the proposed DM method and an approach based on the idea of forward semi-supervised feature selection are able to successfully maximise accuracy on the evaluation dataset, if parameters are tuned appropriately. The greedy forward selection, as well as LSDF, do not succeed in selecting the feature which maximises accuracy on evaluation data.

4.3.2 EMNIST

In the case of EMNIST, accuracies on evaluation data for the first six selected features are presented in Figure 4.3 for different ratios of unlabelled to labelled data. With six features, an good accuracy in comparison to the maximum possible can already be reached, if features are chosen sufficiently. Figure A.3 shows comparisons of methods for different feature counts as well.

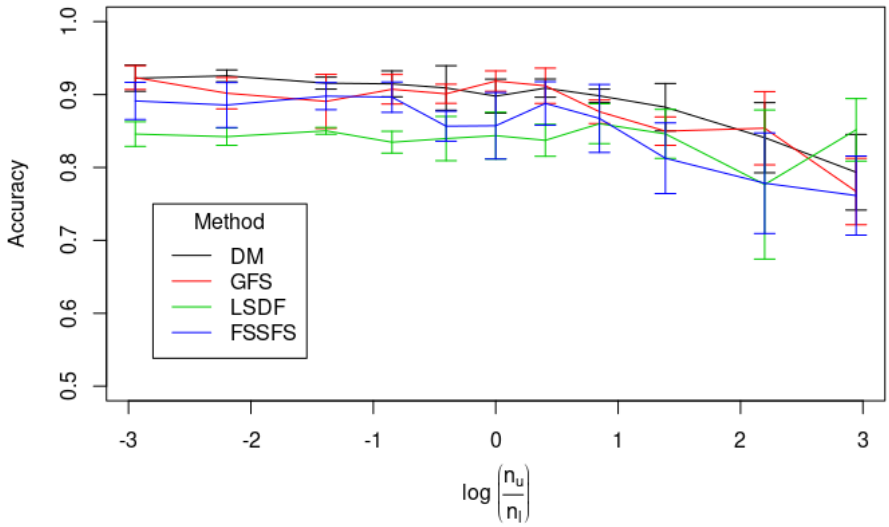


Figure 4.3: Comparison of different methods for EMNIST data with different ratios n_u/n_l, for $n_l + n_u = 1000$, for five features. The reported accuracy is the mean over five different random seeds, and the error bars indicate the standard deviation. DM is distribution matching, GFS is greedy forward selection, and FSSFS is forward semi-supervised feature selection. For distribution matching, $\log \beta = 2$.

The mean accuracies on evaluation data are, for distribution matching and for most ratios, above those of the other methods. However, when accounting for uncertainty, it cannot be claimed that the distribution matching method in this particular experiment outperforms greedy forward selection. It does seem to perform somewhat, albeit not always clearly, better than forward semi-supervised feature selection. LSDF, on the other hand, performs noticeably worse for $\log(n_u/n_l) < 0.5$. For small amounts of labelled data, on the right side of the plot, LSDF does however outperform the wrapper methods. Generally, accuracies achieved by wrapper models decrease with smaller amounts of labelled data.

The superiority of LSDF for a small labelled dataset can be explained by the fact that it does not need to train models. LSDF converges much slower to maximum achievable accuracy features, which is the reason why it is outperformed by wrapper methods in Figure 4.3 whenever sufficient labelled data is available. In Figure A.3 it

can be seen how the gap between LSDF and wrapper methods only slowly starts to close with the selection of more features.

4.3.3 Run time comparison

For EMNIST, the run times of all four methods are compared as well. According to Table 4.4, LSDF is the fastest. This is expected, since no classifiers need to be trained. The proposed distribution matching method is slower than greedy forward selection, since it is an extension of greedy forward selection. The added time is for the most part invested in calculating distances between data points. Forward semi-supervised feature selection is by far the slowest, since it runs many iterations of greedy forward selection. Parameters used to obtain these run times are listed in Table B.1.

Table 4.4: The CPU time required by each method for conducting feature selection of 30 EMNIST features (or ranking all EMNIST features in the case of LSDF), on a Ryzen 7 3.6GHz CPU. The parameters are described in Table B.1. The dataset sizes are $n_l = 300$ and $n_u = 700$.

Method	Run time (sec)
Distribution matching	89
Forward semi-supervised feature selection	5365
LSDF	11
Greedy forward selection	62

5. Discussion

In Chapter 4, distribution matching is analysed and compared against other methods. This chapter discusses implications and possible reasons of the results, as well as potential future work.

5.1 Implications of experimental results

Conclusions from the experiments are discussed regarding the distance measures, the ability of distribution matching to solve the given problem statement, and how it compares to other semi-supervised feature selection methods.

5.1.1 Data distance measures

All three distance measures d_x^{minres}, d_x^{cor}, and d_x^{data} perform equally well. They do succeed in assigning distances as expected, but there does not seem to be an advantage of prediction-based d_x^{minres} and d_x^{cor} over data-based d_x^{data}. That d_x^{data} works well shows that, in the considered EMNIST dataset, the dimensionality is not large enough for distance concentration to occur. In different experimental setups however, especially if the dimensionality of the data is very high, d_x^{data} might approach this limitation where d_x^{minres} and d_x^{cor} do not. Further experiments on differently sized and distributed datasets are required to investigate how distance measures compare in such setups.

The quality of predictions for the prediction-based distances can be influenced by the quality of classifiers trained on different S_j. Performance of these classifiers has been checked to be mediocre when the experiment parameters were determined, since choosing a reasonable size k of S_j is highly dependent on dataset and classifier type. A potential negative effect due to this issue is hence avoided in the presented experiments. While the experiments do not state whether d_x^{minres} and d_x^{cor} were used to maximum potential, the selection of S_j was shown to be sufficient to get reasonable distances.

In further investigations, random sampling of subsets S_j could be improved by checking for duplicate S_j across the sampled subsets. Potentially each subset of features

could be checked for accuracy before it is accepted, to avoid noisiness. In [26], Yaslan et al. propose a method for determining relevant random feature subsets based on weighing features using mutual information. This might prove to be an even better option.

A more thorough investigation is still needed into the variance of data patterns that are picked up by different S_j. A future examination of the impact of p, the number of classifiers that are trained on randomly sampled S_j, on the distance measures would also provide additional insights.

5.1.2 Method comparison

The method comparison experiments allow to conclude that distribution matching is able to minimise $\tilde{L}$ on the synthetic data, where this can be directly measured. Greedy forward selection and LSDF, on the other hand, are not able to minimise $\tilde{L}$, while an algorithm based on the principles of forward semi-supervised feature selection in principle would be able to minimise the expected loss.

On EMNIST data, where the comparison with different methods is based on the first six selected features, the proposed distribution matching outperforms LSDF for sufficient amounts of training data. It also appears to be better than forward semi-supervised feature selection, but for this particular experimental setup and dataset, it could not be shown conclusively that the proposed distribution matching outperforms greedy forward selection.

This, however, does not allow the conclusion that distribution matching is generally not better than greedy forward selection. Instead, the result depends strongly on experimental parameters and should be repeated with other datasets and classifiers. The SVM which is used here, for instance, is a strong classifier that is able to achieve high accuracies in prediction even from comparatively uninformative features, potentially rendering feature selection less meaningful. Other binary classifiers should be tested as well. Examples of those are logarithmic regression or K-nearest neighbour, which is used for comparison in [22] and [23], or Naive Bayes, which is also used in [22]. For these classifiers, superior accuracies of forward semi-supervised feature selection, when compared to greedy forward selection, are shown in [22] for at least some datasets. Furthermore, the EMNIST split that is used in this thesis relies on the assumption that digits 1 and 5 are best distinguished from digit 3 based on different features. Thus, results might already look different if other EMNIST digits are chosen. An entirely different dataset should also be used for testing.

5.2 Future work

Beyond the binary classification tasks considered in the previous chapter, the proposed distribution matching method should also be investigated in other contexts.

Firstly, the difference in distribution between labelled and unlabelled data can be defined in various ways. In this thesis, the frequency of digits in EMNIST was changed to emulate a drift in distributions. While the actual value of the digit was not used as a label in these experiments, the approach still is equivalent to changing the distribution of labels. In the future, it might be interesting to investigate how a distribution change within the data affects performance. This could for example be achieved by differentiating between right-tilted digits and left-tilted digits in the EMNIST case.

Secondly, distribution matching can be applied to any modelling task. Multi-class label datasets and regression tasks were not discussed in this thesis, but should be investigated as additional application domains. Multi-class label setups in particular would make a shift in distributions between labelled and unlabelled data easier to create than binary classification setups. For regression problems, [14] and [15] provide Laplacian based methods comparable to the LSDF method used in this thesis. An approach similar to forward semi-supervised feature selection, which is newer and algorithmically more complex, is presented in [2]. It is, like distribution matching, based on training several classifiers on various subsets S_j and, for classification tasks, would also be a good method to compare against.

Additionally, DM only provides a cost function $c(S_j)$ to optimise. While this thesis uses greedy forward selection to find a local optimum of $c(S_j)$, arbitrary other search functions such as backward elimination should be attempted as well.

Since distribution matching is based on the assumption of a well-working distance measure, additional research into distance measures might optimise achievable performance. This does not only include investigation and improvement of the measures presented in this thesis, but entirely new measures might be offering additional advantages.

Departing from the distribution matching approach to semi-supervised feature selection, the idea of the distance measures itself can be extended into several directions as well. This thesis only uses distances defined between data points. Distances between individual features could, however, also be defined from the predictions generated for different S_j. With data point distances as well as feature distances available, the interaction between both could be leveraged to not only improve feature selection, but also to investigate interpretability of machine learning models. Distances between features are a useful tool here, since they allow to express similarities and apply unsupervised

techniques such as feature clustering.

6. Conclusion

In this thesis, a new method called distribution matching has been introduced, analysed and tested against comparable methods. The goal of the method is to approximate the expected loss $\tilde{L}$ on unseen data in a semi-supervised setup, even and especially if the available labelled data is biased. The method is a wrapper feature selection method, which can be tailored to any given machine learning model.

Distribution matching could be shown to successfully minimise the expected loss in a transparent analysis on a synthetic dataset. At the same time, comparable methods were not able to account for the difference in distribution between labelled and unlabelled data. A further experiment was conducted on the more complex EMNIST dataset, but with this experimental setup a clear advantage of distribution matching over greedy forward selection, which ignores unlabelled data, could not be proved. Since the outcome is highly dependent on both the used classifier and the dataset, it should be repeated with other classifiers and datasets before drawing final conclusions about the practicability of distribution matching with large real-world datasets.

The only modelling task considered in this work is binary classification, but distribution matching can equally be applied to regression or multi-label classification tasks. Future work is still needed to explore those opportunities.

Acknowledgements

I would like to express my deepest gratitude to my supervisor and group leader Kai Puolamäki, who has taught me a lot during the past year, guided me through the process of research and writing this thesis, and who was always available with answers whenever I had questions. My gratitude extends to Suyog Chandramouli, who helped out whenever there was a need, and who provided valuable ideas to the process of writing this thesis as well. Furthermore, I must also thank Anton Björklund, who worked alongside me on this project, for his advise and the many helpful technical conversations we had.

Beyond those colleagues who were directly involved with the project, I would like to recognise the rest of the research group Exploratory Data Analysis at the University of Helsinki – namely, Rafael Savvides, Henri Suominen and Emilia Oikarninen – for the time we spent together, the pleasant work environment they provided, the things I have learned from them, and the shared experiences.

The idea of working on feature selection was sparked during a previous research project on modelling air pollutant concentrations by replicating physical simulations. My appreciation extends to meteorologists Leena Järvi and Mona Kurppa, who provided the idea, data, and a lot of guidance during that project.

Finally, I would like to express my gratitude to those people who shaped the Master's Programme in Data Science at the University of Helsinki into what it is today. Without the excellent courses, learning opportunities and environment they provided, none of this would have been possible. In particular, these are Jussi Kangasharju, Antti Honkela, Hannu Toivonen and the involved administrative staff, as well as all professors and teachers.

My work in the research group, including this thesis, was supported by the Academy of Finland (decisions 326280 and 326339) and Helsinki Institute for Information Technology HIIT.

Author's contribution

The presented work was done in collaboration with Kai Puolamäki, Anton Björklund and Suyog Chandramouli as a research project. I have contributed to all parts, including literature review, developing the presented method and designing and conducting experiments, as well as interpreting their results.

Bibliography

[1] H. Barkia, H. Elghazel, and A. Aussem. Semi-supervised feature importance evaluation with ensemble learning. In *2011 IEEE 11th International Conference on Data Mining*, pages 31–40, 2011.

[2] F. Bellal, H. Elghazel, and A. Aussem. A semi-supervised feature ranking method with ensemble learning. *Pattern Recognition Letters*, 33(10):1426 – 1433, 2012.

[3] K. P. Bennett and A. Demiriz. Semi-supervised support vector machines. In *Advances in Neural Information processing systems*, pages 368–374, 1999.

[4] G. Biau and D. M. Mason. High-dimensional p-norms. In *Mathematical Statistics and Limit Theorems*, pages 21–40. Springer, 2015.

[5] M. Binshtok, R. I. Brafman, S. E. Shimony, A. Martin, and C. Boutilier. Computing optimal subsets. In *AAAI*, pages 1231–1236, 2007.

[6] A. Björklund, A. Henelius, E. Oikarinen, K. Kallonen, and K. Puolamäki. Sparse robust regression for explaining classifiers. In *International Conference on Discovery Science*, pages 351–366. Springer, 2019. This reference is for the conference paper, but the pdf is the journal version.

[7] P. Blanchard, D. J. Higham, and N. J. Higham. Accurate computation of the log-sum-exp and softmax functions. *arXiv preprint arXiv:1909.03469*, 2019.

[8] G. Brown, A. Pocock, M.-J. Zhao, and M. Luján. Conditional likelihood maximisation: A unifying framework for information theoretic feature selection. *J. Mach. Learn. Res.*, 13:27–66, Jan. 2012.

[9] R. Caruana and D. Freitag. Greedy attribute selection. In W. W. Cohen and H. Hirsh, editors, *Machine Learning Proceedings 1994*, pages 28 – 36. Morgan Kaufmann, San Francisco (CA), 1994.

[10] J. W. S. M. O. Chapelle, M. Pontil, and V. Vapnik. Feature selection for SVMs. *Proc. Advances in Neural Information Processing Systems (NIPS)*, 12, 2000.

[11] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik. EMNIST: an extension of MNIST to handwritten letters. *CoRR*, abs/1702.05373, 2017.

[12] G. Daniel. *Principles of artificial neural networks*, volume 7. World Scientific, 2013.

[13] F. M. Dekking, C. Kraaikamp, H. P. Lopuhaä, and L. E. Meester. *A Modern Introduction to Probability and Statistics: Understanding why and how*. Springer Science & Business Media, 2005.

[14] G. Doquire and M. Verleysen. Graph laplacian for semi-supervised feature selection in regression problems. In *International Work-Conference on Artificial Neural Networks*, pages 248–255. Springer, 2011.

[15] G. Doquire and M. Verleysen. A graph laplacian based approach to semi-supervised feature selection for regression problems. *Neurocomputing*, 121:5 – 13, 2013. Advances in Artificial Neural Networks and Machine Learning.

[16] E. Gabrilovich and S. Markovitch. Feature generation for text categorization using world knowledge. In *IJCAI*, volume 5, pages 1048–1053, 2005.

[17] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

[18] N. Japkowicz and S. Stephen. The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449, 2002.

[19] T. Li, C. Zhang, and M. Ogihara. A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. *Bioinformatics*, 20(15):2429–2437, 2004.

[20] T. M. Mitchell. Generalization as search. *Artificial Intelligence*, 18(2):203 – 226, 1982.

[21] L. C. Molina, L. Belanche, and A. Nebot. Feature selection algorithms: a survey and experimental evaluation. In *2002 IEEE International Conference on Data Mining, 2002. Proceedings.*, pages 306–313, 2002.

[22] J. Ren, Z. Qiu, W. Fan, H. Cheng, and P. S. Yu. Forward semi-supervised feature selection. In T. Washio, E. Suzuki, K. M. Ting, and A. Inokuchi, editors, *Advances in Knowledge Discovery and Data Mining*, pages 970–976, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[23] R. Sheikhpour, M. A. Sarram, S. Gharaghani, and M. A. Z. Chahooki. A survey on semi-supervised feature selection methods. *Pattern Recognition*, 64:141–158, 2017.

[24] J. A. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300, 1999.

[25] H. Yang and J. Moody. Feature selection based on joint mutual information. In *Proceedings of international ICSC symposium on advances in intelligent data analysis*, pages 22–25. Citeseer, 1999.

[26] Y. Yaslan and Z. Cataltepe. Co-training with relevant random subspaces. *Neurocomputing*, 73(10-12):1652–1661, 2010.

[27] Yongkoo Han, Kisung Park, and Young-Koo Lee. Confident wrapper-type semi-supervised feature selection using an ensemble classifier. In *2011 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC)*, pages 4581–4586, 2011.

[28] J. Zhao, K. Lu, and X. He. Locality sensitive semi-supervised feature selection. *Neurocomputing*, 71(10):1842 – 1849, 2008. Neurocomputing for Vision Research Advances in Blind Signal Processing.

[29] X. Zhu and A. B. Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130, 2009.

[30] X. J. Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2005.

Appendix A. Figures

This chapter contains figures which support the experiments of Chapter 4, but are not included in the main text.

A.1 Distribution matching

The figures in this section are related to Section 4.2.1 and 4.2.2, in which the proposed distribution matching method is analysed experimentally.

A.1.1 Distributions of d_x and w_i

Figure A.1 shows how average distances d_x and weights w_i are distributed for EMNIST data, which is discussed in Section 4.2.1.

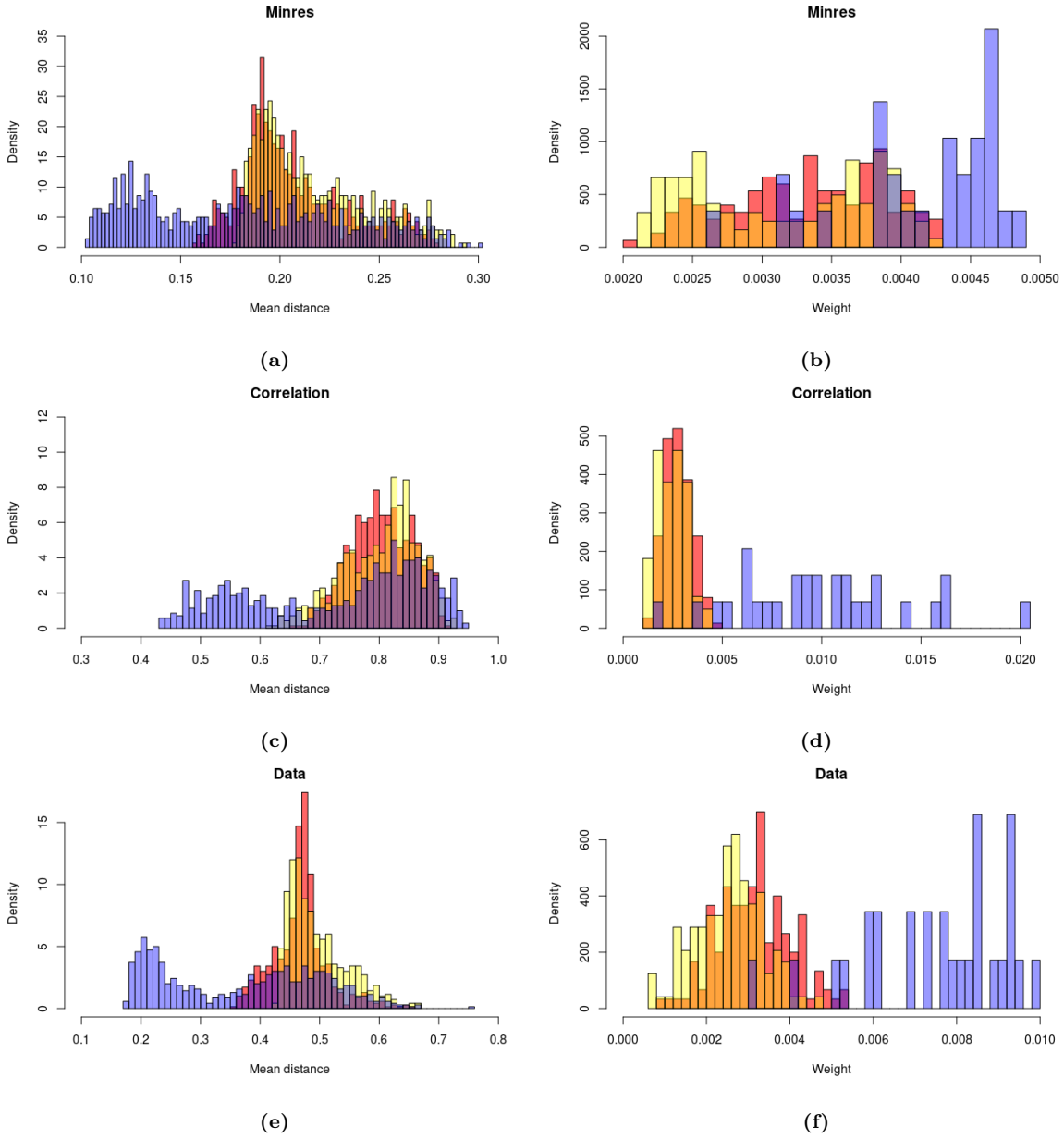
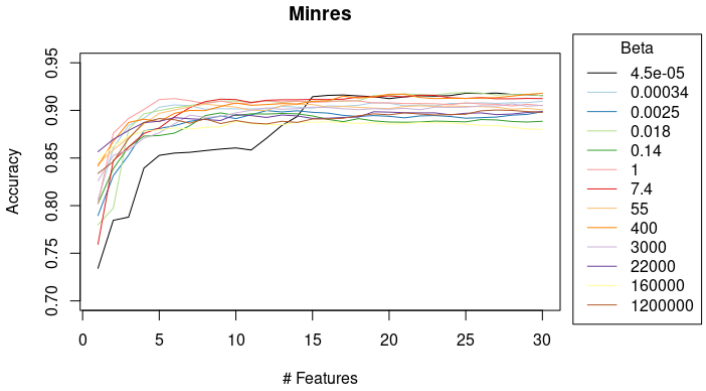


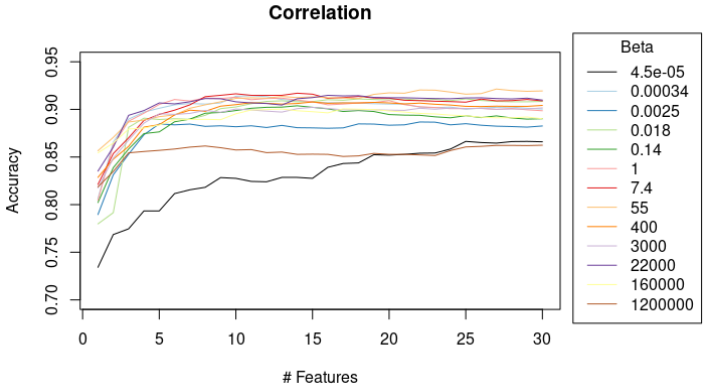
Figure A.1: Comparison of distributions of d_x and w_i for EMNIST digits (Red: 3, Yellow: 5, Blue: 1). Mean distance refers to the average distance of a labelled data point to all unlabelled data points. Rows from top to bottom are calculated with d_x^{minres}, d_x^{cor} and d_x^{data}, respectively. The left column shows the distances, the right column the weights. All experiments use $\log \beta = 2$ and $n_l = 300$, $n_u = 700$.

A.1.2 Feature selection for different β

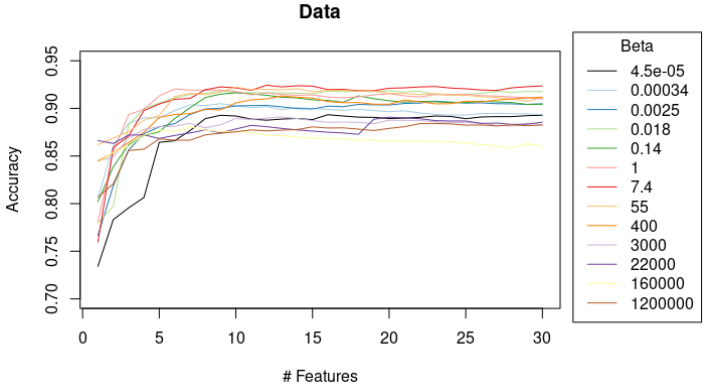
Figure A.2 shows the full feature selection process of the proposed distribution matching method for different distance measures d_x. This is discussed in Section 4.2.2.



(a)



(b)



(c)

Figure A.2: Full feature selection process on EMNIST for different β and distance measures d_x^{minres}, d_x^{cor} and d_x^{data}. Reported here are mean accuracies over different random sub-samples of EMNIST. In all experiments, the number of labelled data points is $n_l = 300$. The β values tested here are exponentials of integers from -5 to 14.

A.2 Method comparison

The Figure A.3 shows how different semi-supervised feature selection methods perform in comparison to each other, for EMNIST data. The topic is discussed in Section 4.3.2.

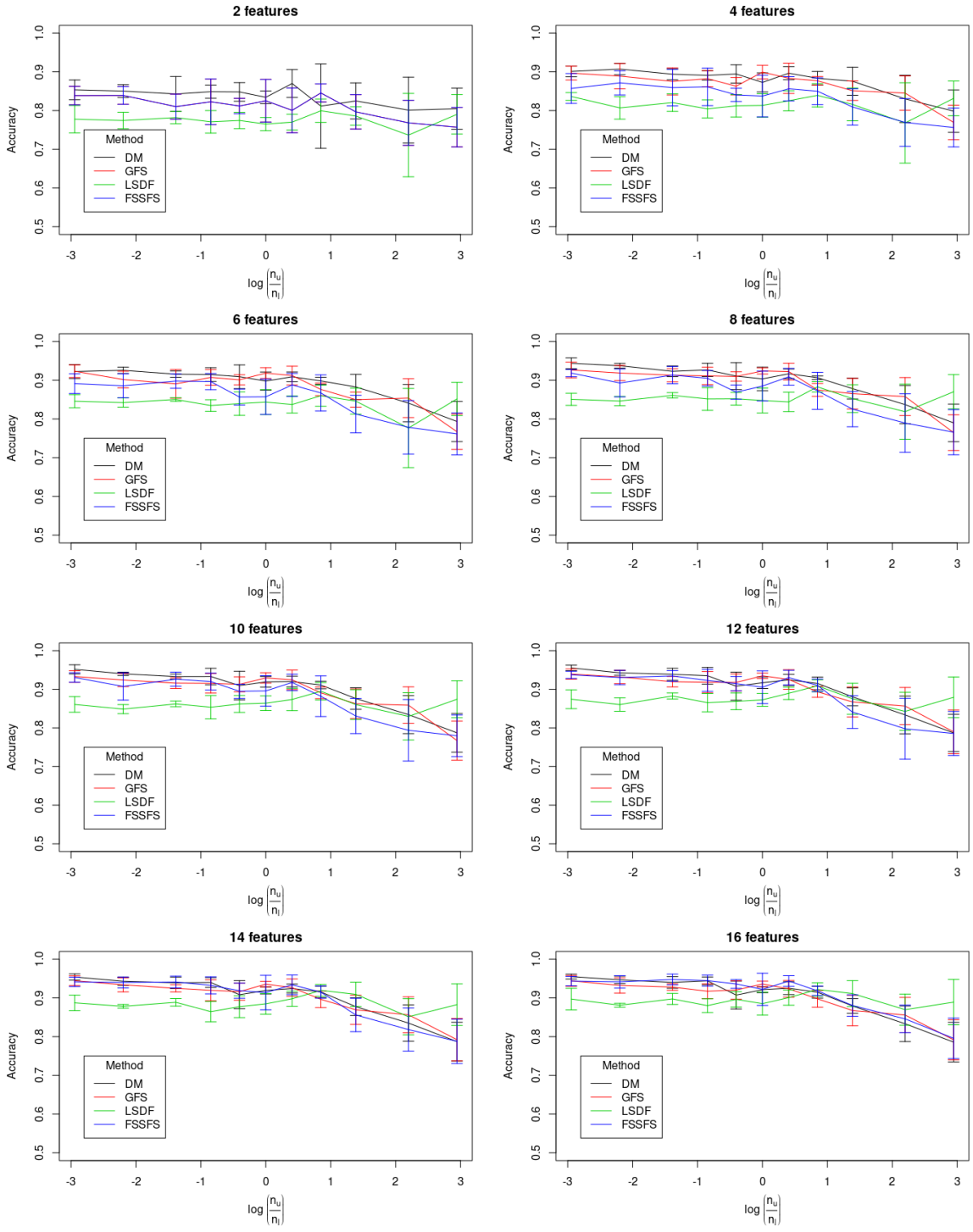


Figure A.3: Accuracy of first 2, 4, 6, 8, 10 and 12 selected EMNIST features for different methods, and different ratios of $\log(n_u/n_l)$. The reported accuracy is the mean over five different random seeds, and the error bars indicate the standard deviation. GFS and FSSFS overlap for two features, since FSSFS uses GFS to select the first two features. Across all ratios, $n_u + n_l = 1000$. DM is the presented distribution matching method. GFS is greedy forward selection and FSSFS is forward semi-supervised feature selection. The distance measure for distribution matching is d_x^{minres}, and $\log \beta = 2$.

Appendix B. Tables

This chapter presents Table B.1, which contains the parameters used for all experiments. Its caption also contains information about why these specific parameters were chosen.

Table B.1: Method parameters. PLTr means percent of the labelled dataset used for training a classifier. The remainder is used for validating the classifier. Parameters j and k are explained in Section 3.3. Parameters γ and K are edge weights and K in K-nearest neighbour, respectively, for LSDF (see [28] for reference). The parameters of forward semi-supervised feature selection written in italics are also taken from the respective paper [22]. Both LSDF and forward semi-supervised feature selection use the default parameters suggested by their authors. An exception to this rule is the $startfn$ parameter, which is reduced to 2 in order to provide a difference to greedy forward selection even for small feature sets. The optimal classifier is defined in equation (4.1). It requires no training, hence PLTr is 0% with the optimal classifier. The SVM used here is the implementation from the R package e1071 with default parameters. Distribution matching for synthetic data allows all possible S_j, which can have size $k = 1$ or 2, hence $k =$Any. A "-" indicates that the method is not applicable.

		EMNIST	Synthetic data
Method	Parameter		
Distribution matching	Classifier	SVM	Optimal classifier
	Measure	Accuracy	Accuracy
	PLTr	66%	0%
	p	200	10
	k	8	Any
Forward semi-supervised feature selection	Classifier	SVM	-
	Measure	Accuracy	-
	PLTr	66%	-
	$startfn$	2	-
	$fnstep$	6	-
	$samplingTimes$	10	-
	$samplingRate$	0.5	-
LSDF	γ	100	100
	K	5	5
Greedy forward selection	Classifier	SVM	Optimal classifier
	Measure	Accuracy	Accuracy
	PLTr	66%	0%